Incorporating Temporal Information in Microblog Retrieval

Craig Willis*
Graduate School of Library & Information Science
University of Illinois at Urbana-Champaign
Urbana-Champaign, IL, USA
willis8@illinois.edu

Richard Medlin
School of Information & Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
rich_medlin@med.unc.edu

Jaime Arguello†
School of Information & Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
jarguello@unc.edu

ABSTRACT

Microblog retrieval is the task of retrieving relevant tweets in response a query. This paper presents our methods and results for the Real-time Ad-hoc Task in the TREC Microblog Track 2012. Our experiments focused on different ways of using temporal information to improve retrieval. Our four runs include three methods that use temporal information and a baseline method that does not. One of our temporal methods favors recent tweets and the other two favor tweets from time periods associated with a high concentration of tweets predicted relevant (potentially, but not necessarily the recent past).

1. INTRODUCTION

Real-time microblog retrieval is the task of retrieving relevant tweets in response to a query. The input to the system is a query Q issued at a particular time t_Q and the output is a ranking of tweets that were published prior to time t_Q.

The School of Information and Library Science at the University of Carolina at Chapel Hill submitted four runs to the Microblog Track's Real-Time Task. None of our runs used *external* or *future* information. That is, the methods used did not use any kind of information that is external to the collection of tweets (e.g., information derived from linked-to webpages) or any kind of information that would not be available at the time the query was issued (e.g., query-expansion terms derived from future tweets).

Our focus was to investigate different ways of using temporal information from tweets predicted relevant in order to improve retrieval. In a general sense, the assumption is that for many queries, the relevant tweets are concentrated in temporal bursts in the past and that identifying these bursts and favoring tweets published during those time periods can improve retrieval.

2. CORPUS PREPROCESSING

The TREC 2012 Microblog Track used the Tweets2011 collection.[1] The original collection consisted of approximately 16 million tweets that were randomly sampled over a period of two weeks between January 24, 2011 to February 8, 2011. Our version of the collection was downloaded using the HTML API on February 21, 2012 and contained only about 12 million tweets. Thus, approximately 4 million tweets present in the original collection were not downloaded either because the Twitter user deleted the tweet since the original collection was created or due to a download error.

Based on the track guidelines, non-English tweets were considered non-relevant *a priori*. Therefore, our main preprocessing step was to remove non-English tweets using the following heuristic. Each tweet was tokenized and every token was issued to ASpell, a freely available open-source spell checker.[2] All tweets for which more than half of its tokens were not found in ASpell were removed from the collection. This procedure removed 65% of all downloaded tweets, leaving a corpus of 4,168,266 tweets.

As previously mentioned, we did not want our runs to use future information. To this end, we constructed 60 different indexes (one per query). Each index contained only the subset of tweets published prior to the query date/time.

All indexes had stopwords removed using a customized stopword list of 189 terms. Stopwords were selected based on part of speech and IDF value. For convenience, IDF values were computed using the *entire* collection. That is, we did *not* generate a different stopword list for each query-specific corpus. One might view this as using future evidence. However, we expected the terms with the lowest IDF value to be stable across the different query-specific corpora and the entire two-week corpus.

3. ALGORITHMS

As previously mentioned, our focus was to explore different ways of incorporating temporal information to improve retrieval. We submitted four runs to the Real-time Microblog Retrieval Task. Our baseline run (Section 3.1) is the only one that does not use temporal information. The recency-based query-expansion approach (Section 3.2), which is a slight modification of the approach from Massoudi *et al.* [3], uses query-expansion the favor recent tweets. The temporal prior approach (Section 3.3), borrowed from Dakka *et al.* [1], conducts an initial retrieval and promotes documents from time periods with a high concentration of top results. Finally, the temporal query-expansion approach (Section 3.4) combines these two previous approaches.

*Work done at the School of Information and Library Science at the University of North Carolina in Chapel Hill.

†Primary author.

[1] `http://trec.nist.gov/data/tweets/`

[2] `http://aspell.net/`

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Incorporating Temporal Information in Microblog Retrieval	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Illinois at Urbana-Champaign,Graduate School of Library & Information Science,Champaign,IL,61820**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Twenty-First Text REtrieval Conference (TREC 2012) held in Gaithersburg, Maryland, November 6-9, 2012. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA). U.S. Government or Federal Rights License

14. ABSTRACT

Microblog retrieval is the task of retrieving relevant tweets in response a query. This paper presents our methods and re- sults for the Real-time Ad-hoc Task in the TREC Microblog Track 2012. Our experiments focused on di erent ways of using temporal information to improve retrieval. Our four runs include three methods that use temporal information and a baseline method that does not. One of our temporal methods favors recent tweets and the other two favor tweets from time periods associated with a high concentration of tweets predicted relevant (potentially, but not necessarily the recent past).

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

3.1 Full Dependence with PRF

Our baseline approach combines Metzler and Croft's Markov Random Field (MRF) retrieval [5] and Indri's pseudo-relevance feedback (PRF) implementation, which is based on Lavrenko's relevance language model [2].

Assuming a uniform document prior, the MRF retrieval model ranks documents according to

$$\begin{aligned} P(D|Q) &\propto P(Q|D)P(D) \\ &\propto P(Q|D) \end{aligned}$$

The query-likelihood component is estimated using Dirichlet-smoothed maximum likelihood estimates

$$P(Q|D) = \prod_{\psi_i \in \Psi(Q)} \left(\frac{tf_{\psi_i;D} + \mu P(\psi_i|C)}{|D| + \mu} \right)^{w_i}, \quad (1)$$

where the ψ_i's are the query features used in Metzler and Croft's full-dependence model [5] (query-term unigram, ordered window, and unordered window features), the w_i's are the weights associated with those features, and μ is a smoothing parameter. We used the default value of $\mu = 2500$ and the w_i's were taken directly from previous work and have been shown to perform well on different tasks and collections [4]. The full-dependence model query ($\mathcal{Q}_{\text{fdm}}$) was implemented using the following Indri query template.

```
#weight(0.80 #combine(unigram query)
        (0.10 #combine(ordered window query)
        (0.10 #combine(unordered window query))
```

Indri's default PRF implementation selects expansion terms with a high probability in a relevance language model derived from an initial retrieval [2]. Given relevance model θ_Q, the probability of candidate expansion term w is given by

$$P(w|\theta_Q) = \frac{1}{\mathcal{Z}} \sum_{D \in \mathcal{R}_{100}} P(w|D)P(Q|D),$$

where $P(w|D)$ is the probability of w given document D's language model, $P(Q|D)$ is the document score (Equation 1), $\mathcal{R}_{100}$ corresponds to the top 100 tweets retrieved for the initial query, and normalizer $\mathcal{Z}$ is computed as

$$\mathcal{Z} = \sum_{D \in \mathcal{R}_{100}} P(Q|D).$$

Because tweets are associated with little text, we decided to extract expansion terms from the top 100 results rather than the top 10, which is a more common heuristic. The expanded relevance model query ($\mathcal{Q}_{\text{rm}}$) was constructed from the top 10 terms with the highest probability in θ_Q using the following Indri query template

$$\#\texttt{weight}(\lambda_1\ w_1\ \lambda_2\ w_2\ ...\ \lambda_{10}\ w_{10}),$$

where $\lambda_i = P(w_i|\theta_Q)$.

The final query was implemented using the following Indri query template

$$\#\texttt{weight}(0.50\ \mathcal{Q}_{\text{fdm}}\ 0.50\ \mathcal{Q}_{\text{rm}})$$

We consider this our baseline approach because it does not use temporal information and denote it as UNCQE (UNC query-expansion).

3.2 Recency-based Query-Expansion

Our recency-based query-expansion approach is a slight modification of the query-expansion method described in Massoudi *et al.* [3]. Candidate expansion terms are scored based on their level of co-occurrence with the original query-terms in *recent* tweets.

The original approach from Massoudi *et al.* [3] scores candidate expansion terms as follows. Let t_Q denote the time query Q was issued and let t_D denote the time document D was published. Candidate expansion term w is scored according to

$$\text{score}(w, Q) = \left(\frac{1}{|Q|} \sum_{q \in Q} \sum_{\{D:q,w \in D\}} e^{-\beta(t_Q - t_D)} \right) \times \log\left(\frac{N}{df_w} \right),$$

where N denotes the number of documents in the collection (which only included documents published before t_Q), df_w denotes the number of documents where w appears, and parameter β controls the amount of temporal decay.

The above equation can be interpreted as follows. The first component considers the average number of documents where term w co-occurs with a query-term $q \in Q$. However, through the function $e^{-\beta(t_Q - t_D)}$, it favors co-occurrences in more *recent* documents. The second component is simply term w's inverse document frequency (IDF) and is added in order to filter near stopwords that have a high co-occurrence with every term (not only the query-terms).

One potential limitation of this approach is that it may favor expansion terms with a disproportionately high co-occurrence with one of the query terms (e.g., "war" in the query "mexico drug war"), but not *all* of the query terms. Our slight modification of the scoring formula above was done in order to favor candidates terms with a high co-occurrence with *all* of the query-terms. To this end, rather than use the arithmetic mean, we used the harmonic mean

$$\left(\frac{1}{|Q|} \sum_{q \in Q} \left(\sum_{\{D:q,w \in D\}} e^{-\beta(t_Q - t_D)} \right)^{-1} \right)^{-1} \times \log\left(\frac{N}{df_w} \right)$$

The recency-biased expanded query ($\mathcal{Q}_{\text{rb}}$) was constructed from the 10 terms with the highest score using the following Indri query template

$$\#\texttt{weight}(\lambda_1\ w_1\ \lambda_2\ w_2\ ...\ \lambda_{10}\ w_{10}),$$

and was combined with the original full-dependence model query as

$$\#\texttt{weight}(0.50\ \mathcal{Q}_{\text{fdm}}\ 0.50\ \mathcal{Q}_{\text{rb}}).$$

This approach favors recent tweets by expanding the query with terms that co-occur with the original query-terms in the most *recent* tweets. Hence, we refer to it as the recency-based query-expansion approach and denote it as UNCRQE.

One potential limitation of this approach is that the most relevant time periods for a particular query may not be the most recent. This is particularly the case if the query concerns recurring events (e.g., "earthquakes"). The next two approaches are aimed to address this issue.

3.3 Temporal Prior

In order to favor tweets from relevant time periods, which may not necessarily be the most recent, we adopted the approach described in Dakka *et al.* [1]. This approach identifies relevant time periods by conducting an initial retrieval, binning the top n results by date/time, and then finally favoring results from the largest bins (i.e., those time periods associated with many of the top n results).

More formally, the approach works as follows. In probabilistic IR, documents are scored according to

$$P(D|Q) \propto P(Q|D)P(D).$$

The approach from Dakka *et al.* [1] assumes that document D can be "decoupled" into a *content* component (D_c) and a *temporal* component (D_t). If we assume that these components are independent given the query, then the scoring function can be written as

$$\begin{aligned} P(D|Q) &= P(c_D, t_D|Q) \\ &= P(c_D|Q)P(t_D|Q) \\ &\propto P(Q|D_c)P(D_c)P(q|D_t)P(D_t). \end{aligned}$$

Furthermore, if we assume a uniform prior for both the content component D_c and temporal component D_t, then

$$P(D|Q) \propto P(Q|D_c)P(Q|D_t).$$

$P(Q|D_c)$ can be estimated using Equation 1 and $P(Q|D_t)$ can be estimated as follows. Given an initial retrieval, the query's temporal profile is generated by binning the top n results into different time periods, using a pre-determined temporal unit (e.g. minute, hour, day, etc.). Note that the number of bins for a particular retrieval would depend on the number of distinct temporal units (e.g., minutes, hours, days) present in top n results. Then, bins are ranked in descending order of size (i.e., number of assigned tweets) and indexed by $i = \{1, ..., T\}$. The first bin ($i = 1$) corresponds to the one with the greatest number of top-n tweets and the last bin ($i = T$) corresponds to the one with the fewest number of top-n tweets. Let the function $\text{bin}(t)$ return the index of the bin corresponding to temporal unit t. The query's temporal profile is defined by

$$P(Q|t) = \lambda e^{-\lambda \text{bin}(t)},$$

where parameter λ (explained in more detail below) controls the amount of decay. Then, for a given document D, published at time t_D

$$P(Q|D_t) = \lambda e^{-\lambda \text{bin}(t_D)}.$$

The final document score is given by

$$P(D|Q) \propto P(Q|D_c) \times \lambda e^{-\lambda \text{bin}(t_D)}. \quad (2)$$

Parameter λ controls the amount of decay. If λ is set to a large value, then the documents published in the time period(s) corresponding to the largest bin(s) are aggressively promoted to the top-ranks. That is, the re-ranked results may substantially differ from the original. Conversely, if λ is set to a small value, then *all* bins are effectively given a similar importance and the re-ranked results will closely resemble the original.

Equation 2 can be viewed as a combination of the full-dependence model with a query-specific temporal prior on document D. Hence, we refer to this approach as the temporal prior approach and denote it as UNCTP.[3]

3.4 Temporal Query-Expansion

The recency-based query-expansion approach described in Section 3.2 scores candidate expansion terms based on their degree of co-occurrence with the original query-terms in *recent* tweets. As previously noted, this may be problematic if the most relevant time periods associated with the query are further in the past.

To address this potential limitation, we extended the recency-based query-expansion approach by using the same binning technique described in the previous section. First, an initial retrieval is produced using a full dependence model ($\mathcal{Q}_{\text{fdm}}$). Then, these results are binned according to the temporal periods reflected in the top n results. Finally, the bins are sorted by size and indexed by $i = \{1, ..., T\}$, where the first bin ($i = 1$) has the greatest number of top-n results and the last bin ($i = T$) has the fewest number of top-n results.

Given a temporal binning of top-n results, the temporal query-expansion approach scores candidate expansion terms according to,

$$\left(\frac{1}{|Q|} \sum_{q \in Q} \left(\sum_{\{D: q, w \in D\}} \lambda e^{-\lambda(\text{bin}(t_D))} \right)^{-1} \right)^{-1} \times \log \left(\frac{N}{df_w} \right),$$

where function $\text{bin}(t_D)$ returns the index of the bin associated with t_D and is in the range $[1, T]$.

The temporally-biased expanded query ($\mathcal{Q}_{\text{tmp}}$) was constructed from the 10 terms with the highest score using the following Indri query template

$$\#\texttt{weight}(\lambda_1\ w_1\ \lambda_2\ w_2\ ...\ \lambda_{10}\ w_{10}),$$

and was combined with the original full dependence model query as follows

$$\#\texttt{weight}(0.50\ \mathcal{Q}_{\text{fdm}}\ 0.50\ \mathcal{Q}_{\text{tmp}}).$$

The temporal query-expansion approach is denoted as UNCTQE.

4. PARAMETER TUNING

Parameters were tuned by maximizing mean average precision (MAP) on the 2011 Microblog Track topics (MB01-MB50).

For the baseline approach (UNCQE), no parameters were tuned. As previously noted, we used the default Indri parameters and we assigned equal weight to the initial (full-dependence model) query and the expanded query. This was the case for all of our query-expansion runs: UNCQE, UNCRQE, and UNCTQE. For the temporal prior approach (UNCTP) and temporal query-expansion approach (UNCTQE), we binned the top $10,000$ results and experimented with hourly and daily binning. Furthermore, we tuned parameter λ across different orders of magnitude from $\lambda = 1$ to $\lambda = 0.0001$. These parameters were set jointly using a grid search. Finally, for the recency-based query-expansion approach (UNCRQE), we tuned the decay parameter β for an approximate tweet half-life of a second, a minute, an hour, and a day.

[3]We admit that this is somewhat of a misnomer because a prior should be query agnostic.

5. RESULTS

Table 1 shows average performance across several metrics for all four of our runs: (1) the baseline approach UNCQE (Section 3.1), (2) the recency based query-expansion approach UNCRQE (Section 3.2), (3) the temporal prior approach UNCTP (Section 3.3), and (4) the temporal query-expansion approach UNCTQE. (Section 3.4). These averages exclude queries MB053, MB068, and MB105, for which the pooling of results yielded no relevant documents. Statistical significance was tested using a paired t-test (paired on queries).

These results show several important trends. The temporal prior approach (UNCTP) had the lowest performance across all metrics and performed significantly worse than the baseline approach (UNCQE) in terms of P@10, P@30, AP, and R-Precision. A possible reason for its poor performance is the following. Different from the rest of our runs, this approach does not expand the original query with new terms. Instead, it simply re-scores the results retrieved by the original (full-dependence model) query. Because tweets are text impoverished, it seems like some form of query and/or document expansion (e.g., using linked-to webpages) is important. Indeed, this approach retrieved fewer results than other three. On average, the other approaches retrieved close to the maximum number of results per query ($10,000$). The temporal prior approach averaged $5,293$ results per query. The temporal prior approach might have performed better in combination with document expansion.

The temporal query-expansion approach (UNCTQE) was the best performing across all metrics. Its improvement over the baseline was marginally significant in terms of MAP ($p = 0.059$) and significant in terms of R-Precision ($p = 0 < 0.05$). This provides modest evidence that exploiting temporal information can improve performance. The temporal query-expansion approach also outperformed the recency-based query-expansion approach (UNCRQE). While it is not shown in Table 1 in order to avoid clutter, its improvement over UNCRQE was significant in terms of AP ($p < 0.05$) and R-Precision ($p < 0.05$). This suggests that the most relevant time periods are not always the most recent. In other words, allowing the model to favor tweets for certain time periods, but not necessarily the most recent, improves performance.

Figure 1 illustrates our per-query performance compared to each query's median- and best-performing system in terms of AP. The queries along the x-axis are sorted in descending order of our run's AP performance. Notice that the lines associated with the median and best performance are different across figures. This is because the ranking of queries is also different. As the figure shows, all of our runs were close to the median performance for each query and far from the best performance for each query. However, these median and best performances include runs that used external or future evidence or both. The UNCQE, UNCTP, UNCRQE, and UNCTQE approaches were above the median performance for 46%, 36%, 41%, and 48% of all queries, respectively.

6. DISCUSSION

Our UNCRQE run used a slight modification of the query expansion method described in Massoudi *et al.,* [3]. The original method computes the degree of co-occurrence between each query term and each candidate expansion term (favoring co-occurrences in recent documents) and then accumulates scores across query terms by using the arithmetic mean. In order to favor candidate terms with a high co-occurrence with *all* the query-terms, we used the *harmonic* mean. Figure 2 compares different means (arithmetic [3], geometric, and harmonic) for P@30 and MAP on last year's queries, which were used for parameter tuning and model selection. The results are shown for different values of β (setting β for an approximate half-life decay of an hour, a minute, and a second). As shown in the figure, the improvement from using the harmonic mean was very small ($< 5\%$) and not likely to be noticeable. However, the improvement was consistent across metrics and values of β, so we decided to substitute the arithmetic mean with the harmonic mean in our final run (UNCRQE).

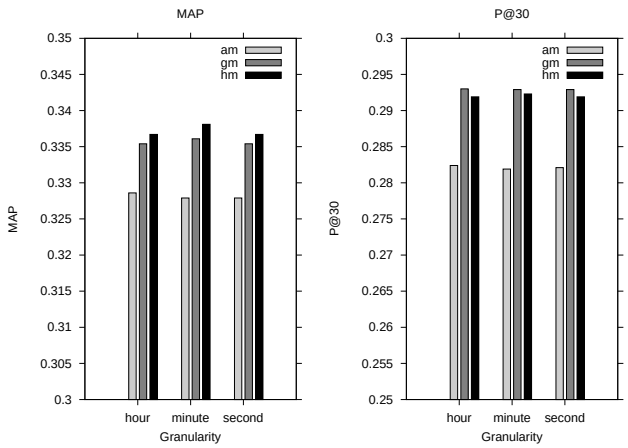


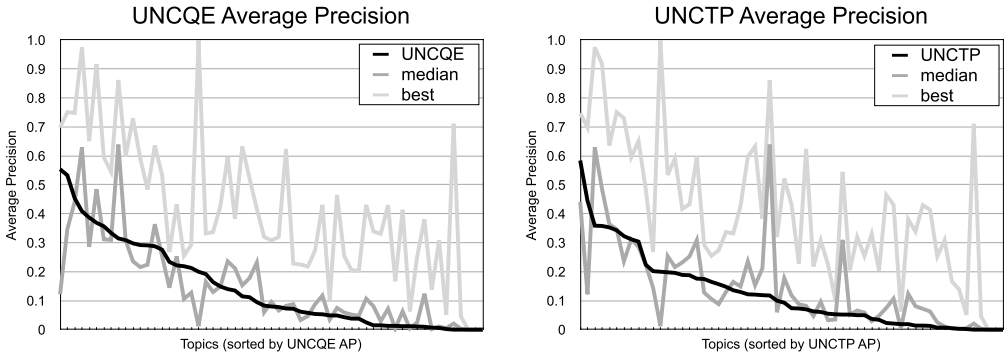
Figure 2: Evaluation of different averaging methods for the approach from Massoudi *et al.,* [3]. The original approach uses the arithmetic mean and our slight modification uses the harmonic mean.

The results presented in the previous section suggest that UNCTP suffered from a lack of query and/or document expansion. We investigated this further by applying the temporal prior re-scoring to the results from UNCQE rather than the results from the full-dependence model query, which uses only the original query terms. We denote this *unofficial* run as UNCQE+TP. These results are provided in Table 2. The lack of query expansion was a major contributor to UNCTP's poor performance. The temporal prior approach improved dramatically by re-scoring the results from UNCQE rather than the results from the full-dependence model query. There was no significant difference between UNCQE and UNCQE+TP.

In the following error analysis, we examine different cases where temporal information improves or deteriorates performance, depending on how it is used. We focus this analysis on the recency-based query-expansion approach from Massoudi *et al.,* [3] (UNCTQE) and the temporal query-expansion approach (UNCRQE). Both methods use query expansion. However, the recency-based approach favors expansion terms from recent tweets and the temporal approach favors expansion terms from relevant busts in the recent (or not-so-recent) past. The relevant bursts are predicted from the temporal distribution of the top-n results from an initial (full-dependence model) retrieval.

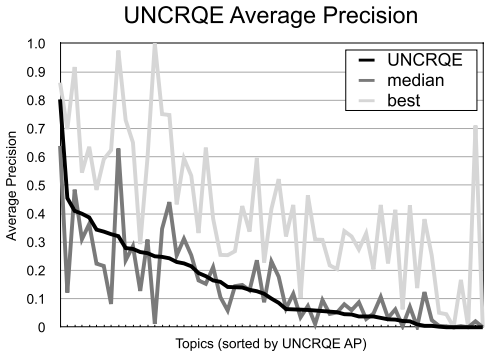
Table 1: Average performance. The percentages correspond to the percent improvement over the baseline UNCQE approach and the ▼ and ▲ denote a statistically significant drop or improvement in performance compared to UNCQE. The best run for each metric is shown in bold.

	UNCQE	UNCTP		UNCRQE		UNCTQE	
P@5	0.271	0.257	-5.26%	0.261	-3.95%	**0.279**	**2.63%**
P@10	0.246	0.214▼	-13.04%	0.248	0.72%	**0.261**	**5.80%**
P@30	0.190	0.164▼	-13.48%	0.192	0.95%	**0.204**	**7.53%**
AP	0.154	0.132▼	-14.08%	0.157	1.97%	**0.173**	**12.33%**
R-Prec	0.187	0.159▼	-15.11%	0.191	1.88%	**0.212▲**	**12.90%**

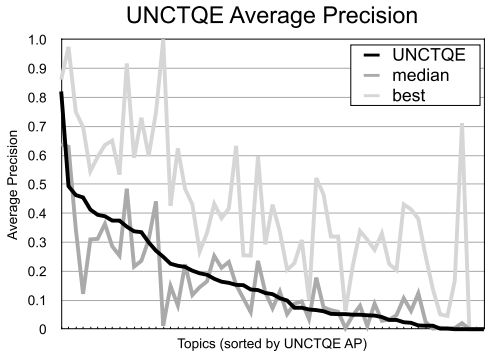


(a) UNCQE

(b) UNCTP



(c) UNCRQE)



(d) UNCTQE

Figure 1: Result in terms of average precision (AP) compared to each query's median and best performance.

Table 2: Average performance. A ▲ denotes a statistically significant improvement in performance compared to UNCTP. UNCQE and UNCQE+TP were statistically indistinguishable.

	UNCQE	UNCTP	UNCQE+TP
P@5	0.271	0.257	0.268
P@10	0.246	0.214	0.248▲
P@30	0.190	0.164	0.188▲
MAP	0.154	0.132	0.155▲
R-Prec	0.187	0.159	0.188▲

Figures 3-5 show the temporal distribution (top) and bin-size distribution (bottom) associated with the top-n initial results (denoted here as the "baseline" run) and the *true* relevant documents for three different topics: Topic 80 (Figure 3), Topic 86 (Figure 4), and Topic 103 (Figure 5). We analyze these topics because UNCRQE and UNCTQE were associated with different levels of performance compared to the UNCQE approach and each other.

In the case of Topic 80 (Figure 3), both UNCRQE and UNCTQE performed considerably better than UNCQE based on AP. UNCRQE outperformed UNCQE by 348.70% and UNCTQE outperformed UNCQE by 199.73%. Figure 3 (top) shows how the *true* relevant documents are concentrated in the most recent tweets (recency in the top figure should be read from right to left). Figure 3 (bottom) also shows how the day associated with the second largest bin had a large concentration of *true* relevant documents (bin-size in the bottom figure should be read from left to right). This helps to explain why both UNCRQE and UNCTQE

performed well for this topic. Furthermore, it can be seen that many relevant documents were published during the day associated with the smallest bin (the one on the far right). This might explain why UNCTQE did not do as well as UNCRQE for this topic.

In the case of Topic 86 (Figure 4), UNCTQE performed better than UNCQE by 20.92% in terms of AP and UNCRQE performed *worse* than UNCQE by 21.38% in terms of AP. Figure 4 (top) shows how the relevant tweets were not the most recent, but were concentrated around 1/24, a week before the query date (1/31). This might explain why UNCRQE failed to generate good expansion terms and performed worse than UNCQE. Figure 4 (bottom) also shows how these relevant bursts were associated with the first, third, and fifth largest bins. That is, many of the *true* relevant documents were published on those days associated with the largest bins. This might explain why UNCTQE outperformed both UNCQE and UNCRQE for this topic.

A similar scenario is depicted for Topic 103 (Figure 5). In this case, UNCTQE outperformed UNCQE by 283.81%, whereas UNCRQE outperformed UNCQE by only 19.81%. Why did UNCTQE do better? As Figure 5 (top) shows, the *true* relevant documents were not concentrated in the most recent tweets. Instead, they were concentrated about a week before the query date. As shown in Figure 5 (bottom), the dates associated with the fourth and sixth largest bins had many *true* relevant documents.

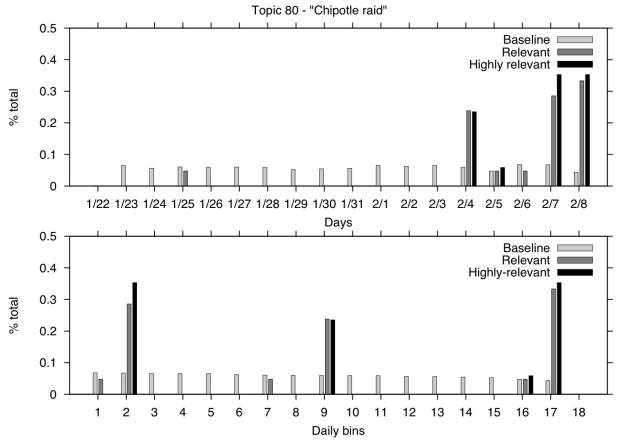


Figure 3: Topic 80: "chipotle raid"

This analysis shows that both query-expansion approaches have pitfalls. The recency-based query-expansion approach suffers when the *true* relevant documents are further in the past and when recent tweets "look" relevant, but are not. Conversely, the temporal query-expansion approach suffers when the initial retrieval fails to identify the bins with the highest concentration of *true* relevant documents.

7. CONCLUSION

We experimented with three different ways of incorporating temporal information to improve ad-hoc microblog retrieval. Our results show two important trends. First, given that tweets are text-impoverished, query-expansion seems to be important. The only method we tested that did not use query-expansion (UNCTP) performed significantly worse than the others. Second, incorporating temporal information can improve retrieval, and it is beneficial to

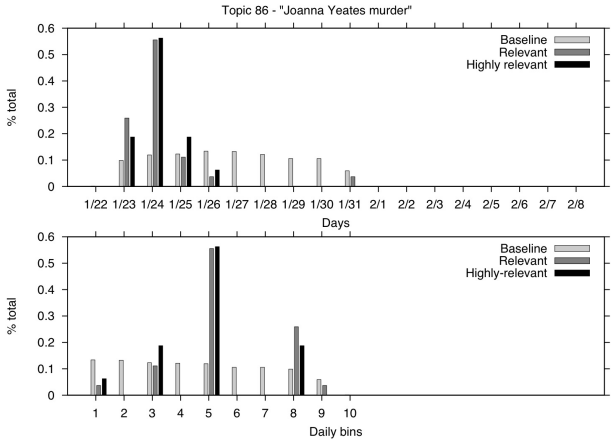


Figure 4: Topic 86: "joanna yeates murder"

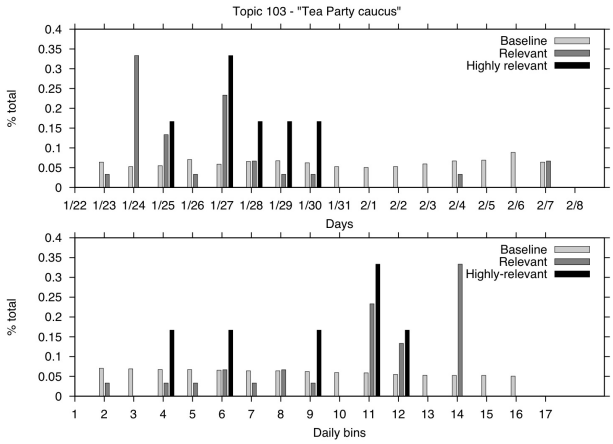


Figure 5: Topic 103: "tea party caucus"

give the model the flexibility of favoring tweets that are not necessarily the most recent. This was the main difference between our UNCTQE and UNCRQE runs, and UNCTQE performed better across all evaluation metrics considered.

8. REFERENCES

[1] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time sensitive queries. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1437–1438. ACM, 2008.

[2] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127. ACM, 2001.

[3] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR '11, pages 362–367. Springer-Verlag, 2011.

[4] D. Metzler. *Beyond Bag of Words: Effectively Modeling Dependence and Features in Information Retrieval*. PhD thesis, University of Massachusetts, 2007.

[5] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479. ACM, 2005.